[0037] It should be noted that some multi-touch input devices, such as direct illumination (DI) setups, can also respond to contact and movement of physical objects. Thus, instead of interacting with his fingers or his hands, a user can interact with a physical object such as stylus, a sheet of paper or a glass. In one embodiment, the present invention can be used to interpret any gestures performed by disposing or moving physical objects across the surface of a multi-touch input device.

[0038] User interfaces that take full advantage of the multi-touch capability still need to be developed, and in particular in the field of gesture recognition. Multi-touch gestures are becoming increasingly important as a modality of interaction in addition to the more common point-and-click interactions. They are also of particular interest to designers of surface computing applications who want to provide an intuitive and natural interface to end-users.

[0039] A key element of effective gesture interaction is the ability to interact without lifting all fingers of the multi-touch surface: it allows a fluid interaction based on continuous gestures. For example, a user can start to resize an object with two fingers, then put a third finger on this object to rotate it with three fingers, lift off a finger to resize it again with two fingers and finally lift off all his/her fingers to release it.

[0040] The recognition of continuous multi-touch gestures is a challenging problem because it requires spatially and temporally segmenting the input data into a sequence of temporally contiguous multi-touch gestures. Dynamically segmenting user's input is a difficult task because is not easy to detect a transition between two distinct multi-touch gestures, i.e. knowing when a multi-touch gesture is ending or beginning. Thus, in current multi-touch recognition systems continuity at the interaction level is ignored. As a result, these systems require explicit breaks between multi-touch gestures, i.e. the user must lift off his/her fingers after each gesture before performing a new gesture.

[0041] It is a therefore a general object of the present invention to provide a method of performing continuous recognition of multi-touch gestures. Simply put, it means that, while passing through a sequence of user inputs, a subset of input data will be extracted and tested against several gesture models to determine the likelihood of a gesture. Once the likelihood exceeds a certain threshold, the corresponding gesture is spotted out and the recognition process start again.

[0042] FIG. **1** shows an exemplary system using the present invention. The system includes at least one multi-touch input device, such as table or a wall, to sense user's input. Each multi-touch input device is associated to a driver or an Application Programming Interface (API) to report data for each contact points with the multi-touch surface, such as touch event type (touch down, lift off, etc.), X and Y coordinates or a unique identifier (ID). The multi-touch gesture recognition engine (MGRE) reads input data from the multi-touch device and interprets them as multi-touch gestures. When a gesture is recognized, the recognition engine generates a gesture event. To receive these gesture events, an application must subscribe to the MGRE, i.e. attach event handlers to a set of gestures. Whenever a gesture event is triggered, the associated event handlers are called with an argument that contains data about the recognized gesture such as its name, parameters or key features.

[0043] The present invention may operate on a single computer or a networked environment, either a local area network (LAN) or a wide area network (WAN), through a network protocol such as TCP/IP or UDP. FIG. **2** illustrates an exemplary distributed architecture using the present invention. In this embodiment, the recognition process is offloaded to a dedicated server. The computing task of multi-touch gesture recognition is then performed by this server, thereby saving computer resources for other tasks such as the tracking of contact points on a FTIR setup or compute-intensive applications.

[0044] FIG. **3** illustrates a multi-touch gesture recognition process in accordance with one embodiment of the invention. The processing method begins with an acquisition block to acquire user inputs: raw data are read from a multi-touch input device. Multi-touch input devices can output various form of data according to their capabilities: list of (X, Y) coordinates, list of tracked points with a unique identifier (ID, X, Y), list of shapes, etc. Thereafter, the raw data can be transformed into a form more suitable for gesture recognition: for example, the system can convert (X, Y) coordinates into a normalized coordinate space to work with device independent coordinates, calculate a list of features (direction, size, velocity, acceleration, boundaries of touched regions, etc.) or track the position of contact points if the input device does not track them. The raw data can also be filtered to reduce noise and avoid false detection: for example, a threshold can be set to remove small contact points. In the following block, the gesture recognition is performed to find which gesture, if any, matches input data. There is no a single algorithm to perform gesture recognition: a variety of techniques or combination of techniques can be used, such as template matching, dictionary lookup, heuristic rules, discrimination nets, neural networks, etc. If the user input is recognized as a gesture input (i.e. a gesture is recognized), the system can either generate a gesture event that will be handled by one or more applications, or directly perform the action corresponding to this gesture: for example, a pinch gesture (see FIG. **12**) can generate keyboard events such as "Control—" that will be interpret as a zoom out command by Google Earth®. This association can also depend on the current context of execution: the same pinch gesture can alternatively be interpreted as a zoom out command if Google Earth® is currently displayed or a command to increase the screen resolution if the desktop is displayed (i.e. no application on screen).

[0045] FIG. **4** illustrates input data acquisition. This task can be separated into two steps: reading raw input from multi-touch input devices and converting raw input data into a common format, e.g. a list of tracker points with a unique identifier and 2D coordinates (ID, X, Y). A multi-touch touch input device can be any combination of hardware, software and telecommunications. For example, an UDP client that can read TUIO ("TUIO: A Protocol for Table-Top Tangible User Interfaces", Kaltenbrunner, M. & Bovermann, T. & Bencina, R. & Costanza, E., Proceedings of the 6th International Workshop on Gesture in Human-Computer Interaction and Simulation, GW 2005) messages is a multi-touch input devices. A computer system with multiple mice connected and a software library that can provide multiple independent mouse cursors, such as Microsoft® MultiPoint SDK or the Multi-Pointer X server (MPX), is also a multi-touch input device. More generally, any tracking system, that is capable of generating data relating at least relating to a succession of absolute or relative positions of multiple objects, can be considered as a multi-touch input device. A number of such tracking systems are available to the person skilled in the art.